

### 3 STRATEGY / ANALYSIS Standards

This section describes naming standards for objects that will be encountered or defined in the Strategy or Analysis phase of application development. All rules established in the Oracle Objects General Rules section must be followed when naming Designer Repository objects, unless specifically excepted by individual object descriptions in this section.

Each of the repository entry's properties is defined as follows:

<b>Mandatory</b>	<b>Required</b>	<b>Optional</b>
Designer will not allow the repository entry to be saved unless that property is filled.	This property should be defined to adequately model the application and be ready for transformation to a design model.	This property need not be filled.

## 3.1 Business Terminology

ANL-01 Business Terminology is used to store acronyms and other business terminology such as that found in the Defense Finance and Accounting Data Model (DFADM) and the Defense Finance and Accounting Process Model (DFAPM). It can also be used to define business terminology or terms that are used in an application system. A comprehensive system glossary can be produced containing these commonly used terms, together with any synonyms and any entities within that application system. The glossary can then be circulated to and agreed by everyone involved in the development project.

Terms entered must not duplicate entity names.

Terminology names can be up to (40) characters in length.

<b>Mandatory</b>	<b>Required</b>	<b>Optional</b>
Name	Description	Comments Notes

Some Business Terminology has Synonyms

<b>Mandatory</b>	<b>Required</b>	<b>Optional</b>
Name		Description Notes

For Business Terminology information run Repository Reports/Global/Elements and Their Application Systems  
Element Type: Business Terminology

## 3.2 Documents

Various types of documents may be produced during application development, to include but not restricted to the following:

- Acquisition Program Baseline (APB)
- Acquisition Strategy
- Analysis of Alternatives Report
- Application Release Checklist (ARC)
- Business Goals
- Business Mission Statement
- Business Objectives for each Goal
- Business Vision Statement
- Capacity Plans
- Data Migration Plan
- Delivery and Acceptance Plan
- DFAS Application Architecture
- Economic Analysis (EA)
- Implementation Plan
- Infrastructure Requirements Description (IRD)
- Life Cycle Cost Estimate (LCCE)
- Military Interdepartmental Purchase Request (MIPR)
- Mission Needs Statement
- Operational Requirements Document (ORD)
- Operations Manual
- Program Managers Charter
- Program Management Plan
- Security Plan
- Software Requirements Description (SRD)
- System Architecture and Requirements Allocation Description (SARAD)
- System Requirements Specification (SRS)
- System Security Authorization Agreement (SSAA)
- Technical Project Plan
- Test and Evaluation Master Plan (TEMP)
- Test Plans
- User Manual

The information that can be specified includes the document name, the location of the file that contains the document, the name of the author, the creation date, the document version number and the current status of the document.

Mandatory	Required	Optional
Name	Type	Ref
Author	Status	Authorization Date
Version	Format	Authorized By
	Source Path	Comment
	Date Document Changed	Notes
	Changed by	
	Document Text	
	Description	

For Business Terminology information run Repository Reports/Global/Elements and Their Application Systems  
Element Type: Business Terminology

The document text field either contains the text of the document or a reference to where the document can be found.

## 3.3 Application Name

### 3.3.1 Naming Convention

ANL-02 Application names will be representative of the Master Development Project it is associated with. Application names will have a maximum of 10 characters.

### 3.3.2 Definition

ANL-03 Each application must have a separate Repository system defined

The repository contains several types of applications. It contains the latest Data Model maintained by DFAS Headquarters and created by numerous applications such as SGL, CEFT, GETBACC, DIFS-R, DPPS, DCAS and DDRS. All applications share common data from the latest PUB. In addition to work areas some applications are targeted for prototype efforts such as SGL\_PROT. Also included in the repository are applications designated as Non Standard Area (NSA). For more information refer to the *DCII Repository Object Creation/Change/Promotion Processes*

Mandatory	Required	Optional
Name	Title	Parent
Datawarehouse? - Indicates whether or not this is a data warehouse application. The value Y assigns an additional, data warehouse specific property to some element types, for example, entity and relational table definitions.	Description	Priorities
	Owner - The application owner is always CASEOWN. Applications are created via requests. The Repository Manager is the only user of Oracle Designer that can create an application. The Repository Manager will grant other repository users access to the application system. For more information on application creation see the <i>DCII Repository Object Creation/Change/Promotion Process</i>	Constraints
	Authority - The name of the person or organization that has authority over this application system. This field has a limit of 10 characters.	Summary
		Notes

ANL-04 Because CASEOWN creates all applications, CASEOWN must populate the Application Properties including the Description. Application information can be sent using MSWORD to the Repository Manager. This information must be provided to the Repository Manager when the application is requested for creation. The Application Description includes the high level inputs, processing and outputs of the application being defined. A high level understanding of the interfaces for the application in context with the DCD, DCW and Non Standard areas must be included. If this information is not available, this should be stated in the notes property of the Application Definition. This information is essential for entry criteria into the analysis phase of a project. The application description should also state who the users of the system are.

## 3.4 Business Units

ANL-05 This object defines the business units of an organization, and also records volumes of entities used at business units and the frequencies of business functions carried out at the business unit. A Business Unit definition will contain a long descriptive name and a short name. Business Unit short names will have a minimum of two (2) characters and a maximum of four (4) characters, two characters is preferred. Business Unit short names are codes which will be derived from the longer, descriptive name of the Business Unit.

### Example:

Long name = Database Services

Short name = DBSV

Mandatory	Required	Optional
Short Name	Parent	Comment
Name	Sequence in Parent	Notes
Role?	Description	Primary Location
		Primary Contact
		Headcount
		Role Cost Rate
		Role Cost Rate Unit (unless a Role Cost Rate is specified)
		Role Responsibilities
		Role Skill 1, 2, 3, 4
		Icon File
		Image File
		Sound File
		Video File
		Execution String
		Note Type
		Note Text
		Note Picture File

Examples of names of currently defined business units include:

- BACC DATA STEWARD
- BACC PROJECT OFFICER
- DCD PROJECT OFFICE
- NON STANDARD DCD PROJECT OFFICE
- SGL MAINTENANCE GROUP
- DCD DENVER GENERAL FUNDS OFFICE
- PRO FORMA ADMINSTRATOR

For information on Business Units run Repository Reports/Function Event Modeling - Function Definition Reports and from Repository Reports/Global - Elements and their Application Systems: element type - business units.

## 3.5 Business Locations

ANL-06 If the Locations of Business Units affect their Business Functions, Locations should be defined.

The location object records information about the geographical location at which business functions are performed. As part of Analysis standards, location information should be added to the repository as required entrance criteria for analysis. If the location of a particular business function is not known this should be recorded as a Problem in Oracle Designer. Certain Assumptions may also be made based on missing information. The lack of information in the Location primary access element should indicate a need for Problems and Assumptions definition within the specific application.

For business units that identify interfaces, the business location should describe in the parent location, the DECC that the interface will process through.

Mandatory	Required	Optional
Name		Comment
Type		Description
		Notes
		Parent Location
		Contact
		Telephone Number
		Address Line 1
		Address Line 2
		Code
		Coordinate 1
		Coordinate 2

Business Unit Planning Items should be defined.

For information on Locations run Repository Reports/Enterprise Modeling /Strategic Considerations.

## 3.6 Business Objectives

ANL-07 Business Objectives will be extracted from the Corporate Strategic Plan and maintained in the Corporate Level Enterprise Model. Business Objective names can be up to forty (40) characters in length.

**Example:** EXPAND TO BE RECOGNIZED NATIONALLY.

Business Objectives records the objectives and aims of an organization; and also statements of business intent, which direct the activities of the organization.

Objectives are more specific targets that help to achieve goals. For example, Goals for the DFAS Community, both functional and technical, include:

- Standardize / Consolidate Operations
- Standardize / Consolidate systems
- Eliminate Problem Disbursements
- Improve Customer Support

Corresponding Objectives for both communities include:

- Reduce Costs through Standardization and Consolidation
- Provide On-line access to financial information at all organization levels
- Ensure users have access to accurate, timely and relevant information
- Assure Security of all DFAS Applications
- Measure and Improve Quality of Service
- Aggressively apply new methods and technologies to improve customer service and reduce operation costs

The relationships between the goals of the technical and functional community are spelled out in the goals and objectives of each application system. Without this information no application should move to the analysis stage. Clear definitions and agreement on the goals of all application development initiatives increases the success rate for the effort. The Strategic Considerations Report available through the Repository Reports includes the following: details of various elements that can be defined at the strategic planning stage of enterprise modeling: assumptions, critical success factors, key performance indicators, objectives and problems.

Mandatory	Required	Optional
Name	Parent Objective (Only for child objectives)	Target Value
Type	Description	Set By
		Unit of Measure (Unless Target Value is specified)
		Planning Horizon
		Date Required
		Ranking
		Achievement Comment
		Comment
		Notes

Objectives need not be associated with Critical Success Factors

Objectives may be associated with Business Functions

Objectives need not be associated with Key Performance Indicators

---

## 3.7 Assumptions

ANL-08      Assumptions should be defined for any functional requirements that have assumptions.

Records and monitors any assumptions that are made about a system; specifically those that may impact the design and implementation of tables. The recording of Assumption in the analysis stage is strongly related to the design of the database. The primary access element - Assumptions is related to the primary access element tables in order to associate between an assumption and the table definitions it influences.

Mandatory	Required	Optional
Name	Description	Comment
Type		Notes
		Made by
		Ranking
		Stability
		Risk



### 3.8 Problems

Records a business event or state that inhibits the progress of the enterprise towards its objective (e.g., personnel disputes, market moves, procedural or statutory changes, unresolved issues or unmade decisions).

ANL-09 All problems associated with the application development initiatives should be defined prior to beginning analysis. Examples of the types of problems that may arise include: Network Performance Issues, Ownership of data issues and functional representation for gathering information. If these problems are stated up front as part of the Strategic Consideration Report, proactive planning is facilitated.

Mandatory	Required	Optional
Name	Parent Problem for child problems	Identified By
Type	Description	Date Identified
		Date Solved
		Opportunities
		Cause Category
		Resolution Benefit
		Comment
		Notes

For information on Problems run Repository Reports/Enterprise Modeling /Strategic Considerations.

## 3.9 Critical Success Factors

Business events, or states, whose failure could seriously hinder the achievement of an objective.

ANL-10 Critical Success Factors must be included in the entry criteria for all application development initiatives. Critical Success Factor Analysis is the identification of a hierarchy of performance measures that lead to identification of critical factors and issues that will determine a business's success. Critical Success Factors are "things that must happen" if objectives are to be achieved. Critical Success Factors also provide a useful baseline not only for evaluating business performance, but also for evaluating information systems' contribution to the value of the business. Specific information system statements may emerge as Critical Success Factors during analysis.

Mandatory	Required	Optional
Name	Description	Parent
Critical		Relative Priority
		Set By
		Target Value
		Unit of Measure
		External Dependency
		Date Required
		Achievement Comment
		Comment
		Notes

For information on Critical Success Factors run Repository Reports/Enterprise Modeling /Strategic Considerations.

### 3.10 Key Performance Indicators

ANL-11      Defines indicators that can be used to quantify or monitor the progress that is made toward achieving a set of business objectives.

Mandatory	Required	Optional
Name	Parent KPI	Comment is not required
	Responsibility	Description is required
	Measured By	Notes are not required
	Set By	
	Target Value	
	Unit of Measure	

For information on Key Performance Indicators run Repository Reports/Enterprise Modeling /Strategic Considerations.

Deliverables, schedules, milestones and other tracking mechanisms will be assigned to each Critical Success Factor and will be used to determine progress. Key Performance Indicator (KPI) names are nouns or nominative phrases of up to forty (40) characters.

**Example:** OPEN 5 NEW OFFICES IN NEW STATES.

---

## 3.11 Business Functions

### 3.11.1 Business Function Modeling

ANL-12 A business function model provides an overview of all the necessary and desired functionality of the business required to achieve the business objectives. The business functions describe not only what the business does, but also what it should do. The scope of investigation is a business area, and focus is on WHAT the business had to do to meet its objectives. There is no distinction between manual functions and functions that need to be supported by the computer system. A sound business function model remains valid as long as the business objectives do not change. Any change in the way the business is run (the how), should not affect the business functions model.

### 3.11.2 System Function Modeling

ANL-13 A system function model focuses on the functionality that must be implemented or supported by the computer system. This includes functions which:

- completely implement business functions
- provide support for selected steps of business functions
- provide different groups of users with different ways of performing business functions (for example and infrequent users' function and a fast-path function for regular users)
- support the way users will work with the system or to help them achieve their business functions, such as functions to support flow of work between users, like work flow queuing and management functions
- required to help administer the operation of the system such as:
- perform error logging
- administer or control system restart and recovery
- report shortage of resources, such as database space
- monitor the integrity of data
- control the flow of processing through the system, such as:
  - \* administer and support the proposed technical architecture, such as network message transmission, routing, and reception
  - \* provide processing workload scheduling: for example batch queuing, sequencing, and monitoring functions for reports and large overnight jobs
  - \* provide real time processing flow and queuing mechanisms
- distribute data across geographically dispersed systems

### 3.11.3 Business versus System Function Modeling

ANL-14 Because of the technique of functional decomposition is the same in both modeling approaches the word function should be used to describe both business and system functions. The need for both business and system function models does not mean you will always end up with two separate function hierarchies. It is very common to have one function hierarchy with the higher levels representing business functions and the lower levels representing system functions. In general, "the what" (business functions), as opposed to "the how" (system functions), should only be taken down two or three levels of decomposition within a function hierarchy.

A system function is a function that has been sufficiently decomposed and may be implemented in the application.

### 3.11.4 Cross-checking with the Data Model

ANL-15 Often, in Business Requirements Definition, only the functional outlines, (for example, the top layers of the function hierarchy), are described. Then a lot of time is spent on creating the entity relationship model. However, it is impossible to create a data model without at least an implicit concept of the functions involved. For this reason, it is essential to start making the functions explicit as early as possible and not to postpone describing the functions to the last few days of requirements modeling. Many discussions on the data model are actually discussions on the functionality and vice versa.

Mandatory	Required	Optional
Label	Master Application	Notes
Short Definition	Master Function	
	Parent Function (only for child functions)	
	Sequence in Parent (only for child functions)	
	Elementary?	
	Frequency	
	Frequency Unit	
	Response Needed	
	Description	

Short Definition should consist of simple Verb-Noun phrases, for example

- Process Vendor EFT Data
- Perform Vendor Input Data Edit
- Suspend Transaction
- Validate BACC Interdependencies / Patterns
- Post JV Transaction Details
- Enter Journal Voucher Details.

### 3.11.5 Functions

ANL-16 The notes property for functions is used for the following information:

Enter notes of a technical nature that are important for the persons involved in the next stages in the system development process.

ANL-17 Do not enter business rules here. Business rules should be expressed as separate functions.

ANL-18 Use Notes to maintain the change history for the particular function.

Mandatory	Required	Optional
Function Type (for elementary functions)	Master Application	Notes
Intention to automate	BPR Generate? (If BPR analysis is to be done.)	Measured Time 1, 2, 3
	Person Cost Rate / Unit	Measured Time Unit
	Overhead Cost Rate / Unit	Quality Percentage
	Total Cost Rate / Unit	Value Added
	Additional Cost Rate / Unit	Value Chain Analysis
	Prior Delay Time / Unit	Competitive Index Value
	Work Time / Unit	Competitive Index
	Quality Check Time / Unit	Resource Required 1,2,3,4
	Post Delay Time / Unit	Problem Areas
	Total Time / Unit	Simple Rules
		Entity Notes
		Icon File
		Image File
		Sound File

### 3.11.6 Function Hierarchy Diagrammer

- ANL-19 Functions must be decomposed to an elementary level.
- ANL-20 All functions on the same level of decomposition must have the same level of abstraction, to the extent possible.
- ANL-21 Decomposed functions must always be fully covered by the functions on the next lower level.
- ANL-22 Use Overnight, Immediate, or leave blank. Attention: This field is used by the Designer utility that creates module entries based on function definitions.
- ANL-23 For every entity, there must be at least one function that allows Create, Retrieve, or Delete (the latter for the sake of completeness).
- ANL-24 Functions should have no overlapping functionality that is not covered by an explicit common function.
- ANL-25 On the lowest level, functions should easily be classified as manual, clerical, computer, or other mechanisms, but no combinations.
- ANL-26 The function hierarchy should be well balanced; that is, all top-level functions should be decomposed to approximately the same number of levels.
- ANL-27 Do not use more than ten subordinate functions per parent function; aim for six.
- ANL-28 Set check box Elementary to checked, only if the function, when executed, always fully succeeds or, if not successful, fully removes its trails. Attention: Elementary functions may be, but rarely are, decomposed.
- ANL-29 Every function must be written in a clear phrase, local to the business, avoiding ambiguity as much as possible.
- ANL-30 The function should not specify the “how”, but the “what” of the business. Use the current “how” of the function explicitly as an example in the function description.
- ANL-31 Function definitions should have the main structure, <verb> [<adjective>] <noun>, or, depending on the language in use, [<adjective>] <noun> <verb>.
- ANL-32 Avoid underscores, punctuation marks, and symbols in the function definition. Use active voice.
- ANL-33 Specify frequency using the smallest possible unit that will form a regular basis; for example, use 800/MONTH rather than 10,000/YEAR if the function is likely to happen every now and then. Use 10,000/YEAR if a peak is likely.
- ANL-34 Functions that have the Archive check box checked should usually have the Delete check box checked as well.

### 3.11.7 Common Functions

- ANL-35 All Common Functions must include the master function label as part of the short definition.

**Example:**

**Label:** M11

**Definition:** Interview the customer (CAM11)

Where CAM11 is the master function “Conduct Interview”.

---

## 3.12 Business Process

- ANL-36 Business processes will be captured using the Process Modeler and the Function Hierarchy Diagrammer.
- ANL-37 All the highest level Business Processes have been identified and entered into the process model. Standard industry descriptions have been, and will continue to be used to identify business processes.
- ANL-38 As additional process models are developed, they will be appended to the appropriate higher level process in the Enterprise Model. These extensions of the Enterprise Model may not exceed two additional levels of business processes before functional decomposition begins.

### 3.12.1 Business Process Label

- ANL-39 At the highest two levels of the enterprise, the business process labels have a single alpha character and have already been established. All subsequent levels of business processes will be composed of three (3) alpha characters where the first of the three characters is the name of the level two parent.

**Example:**

CAM is the label for the Customer Account Maintenance process, which is a child of the level two process 'C' Customer Services.

Labels are automatically displayed in FHD's to help identify processes and functions hierarchic relationships.

### 3.12.2 Business Process short definition

The short definition of a business process is always a noun or nominative phrase.

**Example 1:**

Accounting and Fixed Assets are standard high level processes;

Accounts Payable, Accounts Receivable and General Ledger are processes which are functions of the higher level process, Accounting;

Asset Acquisition, Asset Capitalization, Asset Depreciation and Asset Retirement are processes, which are functions of the higher level process, Fixed Assets.

**Example 2:**

Planning, Customer Service and Marketing are standard high level processes;

Customer Account Maintenance and Collections are processes, which are functions of the higher level process, Customer Service.



## 3.13 Events

An **Event** is a named action, which initiates the execution of a business process. Event names must consist of a verb and an object and are restricted to thirty (30) characters.

### Examples:

RECEIVE INVOICE is an example of a trigger that is an initiating event for an Accounts Payable process.

VENDOR PAID is an example of an outcome event for an Accounts Payable process.

ANL-40 Each Business Process Diagram should both begin and end with a customer Event.

### 3.13.1 Event Definition

In the Repository Object Navigator, it is possible to enter basic information about an event. Usually, all users very easily understand events. Events often form a good starting point for initially generating and identifying functions. Events can be used to group functions logically. Events function as the nodes in the function network.

The Forms Generator does not use event information.

ANL-41 Create events only if the event adds substantially to the user's understanding of the functions of the system and its structure.

ANL-42 Always use explicit events to document the triggering of the following classes of business rules:

- restricted relationship rules
- other entity rules
- change event rules
- data operation rules

Mandatory	Required	Optional
Name	Type	On Condition
	Description	Date / Time
	Frequency	Entity
	Frequency Unit	Attribute
		System Description
		Notes
		Icon File
		Image File
		Sound File
		Video File
		Execution String

---

## 3.14 Business Function Label

All business function labels are limited to ten (10) characters.

Labels are a concatenation of the label from the parent business process, and from one (1) to seven (7) numeric characters. The digits, which identify the level of functional decomposition within the business process are positional and begin with one (1). This scheme supports a maximum of seven (7) levels of functional decomposition of a business process.

ANL-43      Use of alpha characters in the positional parameters will be permitted only on an exception basis with approval of the System Architecture Committee.

### Examples:

#### **CAM1:** Establish New Account

Where:

C = Customer Service (*primary* Business Process)

CAM = Account Maintenance (Business Process which is also a function of the *primary* business process Customer Service)

CAM1 = Establish New Account (First, First Level Function of the process CAM Account Maintenance)

#### **CAM11:** Conduct interview

Where:

C = Customer Service (*primary* Business Process)

CAM = Account Maintenance (Business Process which is also a function of the *primary* business process Customer Service)

CAM1 = Establish New Account (First, First Level Function in the process CAM Account Maintenance)

CAM11 = Conduct interview (First, Second Level Function in the process CAM1 Establish New Account)

#### **CAM12:** Record applicant data

Where:

C = Customer Service (*primary* Business Process)

CAM = Customer Account Maintenance (Business Process which is also a function of the *primary* business process Customer Service)

CAM1 = Establish New Account (First, First Level Function in the process CAM Account Maintenance)

CAM12 = Record applicant data (Second, Second Level Function in the process CAM1 Establish New Account)

---

## 3.15 Business Function Short Definition

ANL-44      Business Functions will have a short definition consisting of a verb and an object. Always describe the function starting with a verb. If you don't it will be confused with a process. The short definition can consist of as few as two words, or it can be as long as a sentence. It should clearly describe the complete activity carried out within the scope of that function.

**Examples:** Establish New Account, Revise Existing Contract

## 3.16 Datastores

ANL-45 A datastore must have a name and an id.

ANL-46 Datastore names describe the contents of the datastore and are restricted to thirty (30) characters. Datastore IDs are limited to ten (10) characters.

### Example:

**CUSTOMER SERVICE INFORMATION** is the name of a datastore that contains files, tables, paper forms, etc. related to Customer Service.

**CSI** is the ID for the datastore.

**Data** is the type for the datastore.

ANL-47 There should be at least one Datastore in each BPD

Mandatory	Required	Optional
ID	Description	Type
Name	Business Unit	Comment
BPR Use in Generation?	Function	Notes
	If BPR Analysis is to be done:	Store Type
	Person Cost Rate / Unit	Frequency / Unit
	Overhead Cost Rate / Unit	Minimum Volume
	Total Cost Rate / Unit	Average Volume
	Additional Cost Rate / Unit	Quality Percentage
	Prior Delay Time / Unit	Value Added
	Work Time / Unit	Value Chain Analysis
	Quality Check Time / Unit	Resource Required 1, 2, 3, 4
	Post Delay Time / Unit	Problem Areas
	Total Time / Unit	Simple Rules
	Measured Time 1, 2, 3 / Unit	Icon File
		Image File
		Sound File
		Video File
		Execution String
		Note Type
		Note Text
		Note Picture File

## 3.17 Entities

ANL-48 The starting point for all data analysis is the FUNC\_PUB Application maintained by the Data Management Team at DFAS Arlington (DFAS-DTB).

All new Entity names will be defined in accordance with the standards and abbreviations that exist at the time of their definition.

### 3.17.1 Entity Names

ANL-49 Entity Names must be singular nouns or nominative phrases. They must be business-oriented, and will contain one blank space between each term. Entity names may not contain any special characters, prepositions or articles. Use the Abbreviation/Acronym list posted to the PAL to determine the approved abbreviation or acronym to be used in the Entity name. The clear text word(s) or the clear text for the acronym used must be entered into the Description and Full Name properties for the Entity.

- Entity names may not contain the names of *physical* constructs such as "file" or "table" as a qualifier.

**Example:** Use **Customer** not *Customer File* or *Customer Table*.

Entities must not exceed 22 characters including spaces. In constructing Entity names, abbreviations and acronyms must be used from the approved list .is The list is posted to the PAL under "DFAS/DCII Standards and Guidelines."

**Example:** CUST ORD, ORD HEADER, MEAS UNT .

- Entity names may not contain any special characters such as: @, #, \$, %, \*, or /.
- Entity names may not contain any prepositions such as: at, by, for, in, of, to.
- Entity name may not contain any articles such as: a, an, the.

### 3.17.2 Entity Plural

ANL-50 All table definitions will use the singular form of the Entity name. Therefore, the Entity plural will be the same as the singular Entity name. Since Designer automatically creates a plural name, the user will have to change it to the singular.

The plural field for each Entity must be defined at the time an Entity is named (since **Oracle** will use this to create the table name). Some utilities impose a restriction on the length of the Entity plural name. For example if you decide to create journal tables using Designer, the tool will generate a three-character suffix "\_JN".

- Entity Plural must not exceed 22 characters including spaces.

### 3.17.3 Entity Short Name (Alias)

ANL-51 Entity short names are automatically created by Designer, but may be changed by the user. The following applies to user-created short names. An Entity short name is composed of a distinct word or words (6 characters or less) or a concatenation of Entity word fragments. Oracle Designer uses the short name in the generation of names for constraints, foreign keys, and sequences. A user should be able to look at an Entity short name and know to which Entity it refers. Since the Entity short name may be used to create any migrated foreign-key column name, it is important that the short name indicate the Entity from which it came. Use a standard abbreviation if one exists. As a guideline , short names should be a minimum of 3 characters and a maximum of 6.

If an Entity name consists of one word, the short name should be the first 3 to 6 characters, or it can be an approved acronym or abbreviation. If the Entity name consists of two or more words, the short name should be the first letter of each of the words in the name not to exceed 6 characters, or each word can be an approved acronym or abbreviation with no space between them. Resolve vulgar or objectionable words with your data administrator as well.

**Examples:** CUST should be the short name for the Entity CUSTOMER;

ADR should be the short name for the Entity ADDRESS;

PHNUM is a possible short name for the Entity PHONE NUMBER. PN will be the short name automatically created;

ORDHDR is a possible short name for the Entity ORDER HEADER. OH will be the short name automatically created.

### 3.17.4 Intersection Entities.

ANL-52 An Intersection Entity associates two different Entities and resolves a many-to-many Relationship. It is named according to its business functionality, if possible.

**Example:**

If Entity one is named ORDER (ORD) and Entity two is named PRODUCT (PRODT), an Intersection Entity may be created called ORDER ITEM (ORD ITM).

If there are no business terms that will work, the name is simply formed from the names of the Entities that are associated.

**Example:**

CUSTOMER ADDRESS (CUST ADR) describes the intersection of the CUSTOMER (CUST) and ADDRESS (ADR) Entities.

### 3.17.5 Association Entities.

ANL-53 An Association Entity is used to resolve a many-to-many **recursive** Relationship. A recursive relationship is created when instances of an Entity may be related to other instances of that same Entity. It is named for its parent Entity with the word ASSOCIATION appended.

**Example:** CUSTOMER ASSOCIATION (CUST ASOC)

The Association Entity will be related to the Entity whose recursivity it is resolving with two one-to-many Relationships (with the Association Entity on the “many” end of the Relationships).

### 3.17.6 Validation Entities

ANL-54 A Validation Entity (also known as a Reference Entity) is one that contains “lookup” or code information. Validation Entities should be suffixed with a blank space and the word "TYPE" to distinguish them from other types of Entities.

**Example:**

ORDER TYPE (ORD TY)  
CUSTOMER TYPE (CUST TY)

### 3.17.7 Subtypes Entities

---

ANL-55 Subtypes should:

- form a complete set of mutually exclusive classes (a partition of the supertype)
- only represent one classification at a time
- include only the Attributes unique to subtypes: subtypes inherit the Attributes of the supertype automatically
- attach only to Relationships specific to subtypes: subtypes inherit the Relationships of the supertype automatically

### 3.17.8 Supertypes Entities

---

ANL-56 Supertypes should:

- include only those Attributes common to all its subtypes

- take part only in those Relationships that are valid for all its subtypes

### 3.17.9 Functional Requirements (Business) Data Modeling versus Technical Development (System) Data Modeling

The DCII Functional Requirements Data Model resides in Designer application FUNC\_PUB and provides a full and detailed definition of the structure of all the data that the business areas use or generate. The DCII Technical Development Data Model resides in Designer application BASELINE and provides a full and detailed definition of the structure of all the data that the system is to store. This includes data identified within the Functional Requirements Data Model that is also within the scope of the system. Both models should contain identical logical objects such as entities and functions except for differences due to requirements specific to a release.

#### 3.17.10 Cross-checking with Function Model

ANL-57 If one or more Entity Relationship Diagrams are being produced, it is critical that they be developed in conjunction with the Function Model to ensure that all the identified Functions are supported by information within the ERD(s). The usage matrices provide a cross-checking mechanism between entities in the ER Diagram and functions in the Function Model. An entity should be supported by at least one function.

#### 3.17.11 Entities and Attributes

Each non-intersecting Entity or Supertype Entity must have at least one Attribute. See Section 3.19 for more information on

Attribute naming. Subtype Entities must have a least one Attribute not found in the Supertype or at least one Relationship to an Entity not related to the Supertype.

#### 3.17.12 Entity Relationship Diagrams

ANL-58 Every conceptual model must be accompanied by a graphical representation of that model.

---

ANL-59 As a guideline to ERDs, all Attributes should be listed.

As a guideline, on ERDs the Primary unique identifier should be listed

#### 3.17.13 Entity Usages

Used by Business Function - each Entity must be associated with one or more lowest level Business Function. To use any of the Entities, they must be attached to an Elementary Business Function that will be responsible for creating, retrieving, updating, or deleting (CRUD) instances of that Entity. As a guideline, each Entity should have a usage of at least one Elementary Function. At least one of the Create?, Retrieve?, Update?, and Delete? properties must be set to yes.

Mandatory	Optional
Create?	Comments
Retrieve?	
Update?	
Delete?	

Each of the Attributes in the Entity must also be associated with each of the Elementary Business Functions that the Attribute's parent Entity is associated with. As a guideline, each Attribute, should have one or more Elementary Business Function that will be responsible for inserting, retrieving, updating, and nullifying (IRUN) Attributes. The Archive and Other usages should not be used. The easiest way to make sure that the Attributes are associated with the Functions is to use the utility menu item: Create Function Attribute Matrix.

## 3.18 Domains

ANL-60 Domains are used to store common Attribute information

A Domain defines a set of Validation rules, format constraints and other properties that apply to a group of Entity Attributes, columns, Oracle object type Attributes, program data constructs, module arguments or data structure items. If you make a change to a Domain, you can propagate the updates to the associated Columns/Attributes using the Update Columns/Attributes in a Domain utility, available from the Utilities menu. Domains are included in the DCII Functional Requirements Data Model (Designer application FUNC\_PUB) maintained by DFAS-Arlington Data Management. All Domains in the DCD may be considered global. Some application may need to create “application specific domains”.

ANL-61 Guidelines for Defining Domains:

- Any Attribute that ranges over a fixed set of 20 or fewer predefined values should be associated with a Dynamic Domain that describes that set of values.
- Any Attribute that ranges over a fixed set of 21 or more values or that is expected to have its set of values change over time should be placed in a reference Entity so that the values of the Domain can be maintained through the use of a form. NOTE: The limit of 20 values is a rule of thumb and may be exceeded if the list of values is very stable. Performance should also be considered in deciding whether to use a Dynamic Domain or reference Entity.
- Static Domains may only be used for a list of values that will never change. For example, YES/NO, TRUE/FALSE. All other Domains will be dynamic.

The following properties pertain to all Domains:

Mandatory		Optional
Name	Average Attribute Length	Authority
Format	Maximum Attribute Length	Unit of Measure
Datatype	Attribute Decimal Places	Derivation
Average Column Length	Description	Validation Rules
Maximum Column Length	Element Steward	Default
Column Decimal Places (for numerics)		Null Value
Dynamic List? –When set to Yes and the display datatype for a column is Poplist or Text List, the list of allowable values associated with the domain will be dynamic (i.e. based on a SELECT statement) rather than hard-coded in the client application		Subset Of
.		Notes

The following properties pertain to Domains with Allowable Values:

Mandatory	Optional
Values	Abbreviation
High Value	Notes
Meaning	
Display Sequence	
Description	

Some Domains have Sub Domains - standards are the same as for Domains

For information on Domains run Repository Reports/Entity Relationship Modeling - Attributes in a Domain Report and Entity Definition Report From Repository Reports/Global - Domain Definition Report



---

**ANL-62** Attributes which have the same characteristics and the same set of values will use the same Domain.

## 3.19 Attributes

An Attribute is any detail, which serves to identify, qualify, quantify, classify, express the state of, or otherwise describe properties of an Entity. Each occurrence of an Attribute within an Entity has one and only one value.

Attributes are classified in the repository by functionality and depicted on Entity Relationship Diagrams with the following symbols:

<u>Symbol</u>	<u>Description</u>
#	Primary Unique Identifier (UID)
*	Mandatory Attribute (may not be null)
O	Optional Attribute (may be null)

ANL-63 Attribute names may not exceed 22 characters in length.

ANL-64 Define Attribute names in singular.

ANL-65 Since Attributes are always shown in the context of an owning Entity, do not repeat the name of the Entity as part of the Attribute name.

ANL-66 Attributes can have multi-word names, with the words separated by a space. Attribute names should be as descriptive as possible without being overly convoluted. As much as possible they should be named using terms an end-user will recognize. Abbreviations and acronyms should be used to create the Attribute name. Abbreviations in the Attribute name must be selected from the approved DFAS abbreviation/acronym list that resides on the PAL under "DFAS/DCII Standards and Guidelines".

### Examples:

- BEG DT (Begin Date)
- END DT (End Date)
- DESC TX (Description Text)

ANL-67 Attribute Names must end in an approved Class Word that indicates the type of data that the Attribute represents. Class Words approved for DFAS and DCII use are as follows:

<u>Class Word</u>	<u>Abbreviation</u>	<u>Data Type</u>	<u>Meaning</u>
Amount	AM	NUMBER	A monetary value.
Code	CD	VARCHAR2	A combination of one or more numbers, letters, or special characters substituted for a specific meaning.
Date	DT	DATE	A particular day of the calendar year, identified by its ordinal number within a calendar month within that year.
Day of Year Date	DOY DT	NUMBER	A particular day of a calendar year, identified by its ordinal number within the year. Represented by a Maximum Length of 3 formatted as "ddd".
Identifier	ID	VARCHAR2	A combination of one or more numbers, letters, or special characters which designate a specific object, but which has no readily definable meaning of its own.
Julian Date	JULN DT	NUMBER	An interval of time in days and fractions since January 1, 4713 BC. Supported by Oracle data type of "J".
Number	NBR	VARCHAR2	A combination of one or more numbers, letters, or special characters which designate a specific object, but which has no readily definable meaning of its own. May be used as an alternative to Identifier.
Name	NM	VARCHAR2	A designation of an object expressed in a word or phrase.

Ordinal Date	ODNL DT	VARCHAR2	A particular day of a specific calendar year, identified by its number within the year. Represented by a Maximum Length of 7 and formatted as "yyyyddd".
Quantity	QY	NUMBER	A non-monetary value.
Rate	RT	NUMBER	A quantitative expression that represents the numeric relationship between two measurable units.
Text	TX	VARCHAR2	An unformatted character string, usually in the form of words.
Time	TM	NUMBER	A notation of a specified chronological point within a day.
Volume	VL	NUMBER	A measurement of space occupied by a three-dimensional figure.
Weight	WT	NUMBER	The numeric expression of the force with which an object is attracted toward the Earth and/or other celestial body by gravitation.
Fiscal Year	FY	VARCHAR2	Any yearly accounting period without regard to its relationship to a calendar year.
Program Year	PY	VARCHAR2	Any yearly budgetary period without regard to its relationship to a calendar year.
Dimension	DIM	VARCHAR2	A one dimensional measured linear distance. The data element definition should be: "The dimension (length, width, height, radius, or elevation, etc) of and/or from"

### 3.19.1 Definition

ANL-68 Define indicator Attributes with the words Indicator Code (IND CD) being the last two words in the name. They should be further defined as VARCHAR2 with a link to the YESNO Domain. Indicator Code Attributes should not be used with any other Domain or list of values.

ANL-69 Attributes that are optional should have a short Attribute note that explains the meaning of a null value occurring for that Attribute, if the meaning is different from VALUE UNKNOWN.

ANL-70 Define all VARCHAR2 Attributes that are case insensitive as uppercase.

ANL-71 Define the Attribute, including a statement of purpose, if necessary, in the Description Property.

Attributes must be supported with the following metadata:

Mandatory	Optional	
Full Name	Domain (if derived from a Domain)	Sequence in Entity
Abbreviated Name	Average Length	On Condition
Optional?	Decimal Places (for numerics)	Units
Format	Percent Used – Initial	Default
Maximum Length	Percent Used – Average	Null Value
Description	Derivation	Comment
	Authority	Notes
	Responsible	Sequence in Sort Key
	Validation Rules	Sort Sequence Order

## 3.20 Relationships

ANL-72 Generally, Entities should have at least one Relationship, although there will be occasional Entities that are standalone.

Relationships should always be named since there can be various Relationships between two Entities, and their purposes must be known. There are many different Relationships possible between two Entities; for example, between the Entities PERSON and COMPANY the Relationship “is currently working for” may be of interest, but so is the Relationship “is owner of”. You must give each end of a Relationship a name, an optionality, and a degree (one or many).

### 3.20.1 Optionality and Degree

It is a good practice to convert many-to-many relationships to many-to-one where possible. Many-to-many and one-to-one relationships should be investigated carefully to make sure they are correct

ANL-73 Relationship Names should fit the following template sentence: *Each FROM-ENTITY {must be, may be} Relationship-name {ONE AND ONLY ONE TO-ENTITY (singular), ONE OR MORE TO-ENTITY( plural)}*

ANL-74 “From” Relationships must be supported by the following metadata:

Mandatory	Optional
From Entity	Abbreviation
From Relationship Name	From Maximum Cardinality
From Minimum Cardinality	From In Arc
From Transferable?	From Average
From Description	From Minimum
	From Maximum
	From Notes

“To” Relationships

Mandatory	Optional
To Entity	To Maximum Cardinality
To Relationship Name	To In Arc
To Minimum Cardinality	To Average
To Transferable?	To Minimum
To Description	To Maximum
	To Notes

Relationships should be named so that the diagram can be easily read. To read any Relationship simply and definitively, the following syntax is used:

Where *FROM-ENTITY* is the source Entity of the Relationship, *TO-ENTITY* is the destination end of the Relationship, and *Relationship-name* is the name applied to the Relationship in the direction the Relationship is being read. Note the following rules:

The choice between *must be* and *may be* is determined by the modality (optionality) of the Relationship emanating from the source Entity. A solid line represents *must be* (mandatory) and a dashed line represents *may be* (optional).

The choice between *ONE AND ONLY ONE TO-ENTITY (singular)* and *ONE OR MORE TO-ENTITY plural* is determined, respectively, by the absence or presence of the “crows feet” at the TO-ENTITY end. This is also known as the cardinality of the Relationship.

Since Relationships are always bi-directional, naming a Relationship requires two Relationship names be supplied. Hence, the Relationship must be readable using the above syntactic structure in both directions.

**Examples:**

Each PERSON *may be* **located at** ONE OR MORE ADDRESSES.

Each ADDRESS *must be* **the location for** ONE AND ONLY ONE PERSON.

Each DEPARTMENT *may be* **responsible for** ONE OR MORE EMPLOYEES.

Each EMPLOYEE *must be* **assigned to** ONE AND ONLY ONE DEPARTMENT.

Whenever multiple Relationships are created into a single Entity, from the same parent Entity, the Relationship labels **must be** unique to prevent duplicate index names from being created by the Database Transformer. Otherwise, the indexes will not be generated during “DDL Generation”

ANL-75 Do not use weak Relationship names, such as “associated with”, or “related to”. Remember that Relationships are documenting business rules and will be used to explain the model to end users. Whenever possible, use business terminology. For example, the correct way to document a person and their address would be:

A PERSON *may be* **located at** one or more ADDRESS.

**NOT**

A PERSON *may be* **associated with** one or more ADDRESS.

Some useful pairs of Relationship names are:

Identifying -	Identified by
Defining -	Defined by
Maintaining -	Maintained by
Creating -	Created by
Servicing -	Serviced by
Sponsoring -	Sponsored by
Specifying -	Specified by
Citing -	Cited by
Detailing -	Detailed by
Describing -	Described by
Relating -	Related by
Classifying -	Classified by
Used to classify -	Classified by
Using -	Used by
Containing -	Comprised of
Authorizing -	Authorized by
Transferring -	Transferred by

Note: where the above are marked with an asterisk \*, one should only use these as a last resort. For example, owned by should only be used as a relationship name when the relationship means legal ownership.

Some of the above names imply the role of a person or an organization.